

then step 702 flows to step 704 which determines that no mouse has been selected, and then step 704 will use the custom button specification table 706 provided by the editor application program. The custom table 706 includes alternate button definitions for the buttons B1, B2, B3, B4, B4and B10, as is shown in FIG. 7C. Step 704 of FIG. 7A calls the driver program 600. The driver program 600 of FIG. 6A then flows to step 601 which determines that no overlay input has occurred and then flows through steps 602 through 608 defining the touch overlay button wires and the LCD display button wires. Then in step 610, since there is a button specification table available for the current application program, namely the table 706 of FIG. 7A, step 610 flows to step 614. Step 614 in FIG. 6A determines that the current application program has a button specification table and then it gets the table 706 and sets the button images and the button values equal to the values provided in the custom table 706 as shown in FIG. 7A. The custom table's values are then loaded into the partition 650 of memory 138 by step 614. Then in step 616 of FIG. 6A, the button display areas of the button panel 17 for the LCD display 15 are painted with the images specified in the custom table 706. Then the button panel driver program of FIG. 6A flows to the return step 634, which returns to step 704 of the editor program of FIG. 7A. The flow diagram of FIG. 7A then flows to step 708 which continues the editor application.

During the course of continuing the editor application in FIG. 7A, if the user elects the mouse input state, then step 710 goes to step 702 which determines whether a mouse input has been selected or deselected and then correspondingly either the default button specification 612 or the custom table specification 706 or another user supplied button specification table will be loaded into the button specification table partition 650 by the button panel driver program 600.

Thus it is seen that the touch sensitive buttons for a digitizing display can be dynamically labeled and relabeled. The invention allows the simultaneous touch inputs or simultaneous stylus inputs both the button panel 17 and the main panel 19. Shift buttons and mouse buttons can be labeled according to their functions, the buttons can be dynamically relabeled by application programs or by the user, multiple buttons can be simultaneously depressed and this can be done with existing programs which do not provide any information on customized button layouts. The user is provided with feedback every time a button is pressed and buttons can be used for designating other functions such as system functions of displaying the time, turning off the main screen, selecting a printer, or other system functions.

FIG. 6B shows a modified sequence of operational steps for the button panel driver program 600. FIG. 6B inserts additional steps between the existing steps 618 and 630, to detect multiple shift buttons which are depressed in a column of buttons defined in the button panel 17. This can be performed where only a single column of buttons have been defined; either all of the buttons are defined for B1, B2, B3, B4, B4or alternately all of the buttons are defined only for the column for buttons B6, B7, B8, B9 and B10, as is shown in FIG. 5. In FIG. 6B, existing step 616 receives a touch or stylus signal from the overlay. Then the step 620 determines if only a single column of buttons has been defined in the button panel 17; then a check can be made for multiple touch or stylus signals within the button panel 17. Otherwise the flow goes to the existing step 630. Step 620 then flows to step 622 which starts a loop for an index i from 1 to 5, for in this example five buttons being defined as a maximum number of buttons in a column. If a signal wire

V(Bi) for horizontal wires and a vertical wire U(Bi) signals are received, then step 622 outputs the button VALUE (Bi) from the partition 650 and highlights each selected button for user feedback. Then step 622 flows to step 624 which returns to the calling program. The modified steps of FIG. 6B allow multiple buttons in a column in the button panel 17 to be detected for either simultaneous touch or simultaneous stylus inputs.

Although a specific embodiment of the invention has been disclosed, it will be understood by those having skill in the art that changes can be made to that specific embodiment without departing from the spirit and the scope of the invention.

What is claimed is:

1. A data processing system, a digitizing display for inputting two simultaneous touch inputs, comprising:

a display coupled to a computer, having a display surface partitioned into a main panel display region and a button panel display region;

a touch overlay coupled to said computer, having a touch input surface partitioned into a main panel touch region juxtaposed with said main panel display region, and partitioned into a button panel to each region juxtaposed with said button panel display region;

means for electrically isolating the main panel and button regions in the display and touch overlay;

a button panel driver program stored in said computer, for defining a button touch area in said button panel touch region in accordance with a button specification table, for outputting a button value when said button touch area receives a touch input;

said button panel driver program defining a button display area in said button panel display region, juxtaposed with said button touch area, said button display area for displaying a button indicium representing said button value and identifying the location of said button touch area;

said computer receiving a first touch input in said main panel touch region and a simultaneous second touch input in said button touch area and in response thereto, outputting a panel value and said button value, the multiple, simultaneous panel and button values being separately identified in a touch circuit;

said computer storing a default button specification table describing a plurality of button touch areas and corresponding button display areas in the event a button specification table is not available;

said button panel driver program using said default table to reconfigure said plurality of button touch areas in said touch overlay and said corresponding button display areas in said display;

an application program stored in said computer, having an alternate button specification table describing a plurality of alternate button touch areas and corresponding alternate button display areas for selection by a user; and

said button panel driver program using said alternate table to configure said plurality alternate button touch areas in said touch overlay and said corresponding alternate button display areas in said display when selected by the user.

2. The system of claim 1, which further comprises:

said touch overlay including both a touch input detection means and a radiative stylus pickup means.

3. The system of claim 2, which further comprises: